

Attorney Docket: T2147-908627  
Application No.: 10/627,976

**AMENDMENT TO THE CLAIMS:**

The following listing of claims will replace all prior versions of claims in the application:

Claims 1 – 20 (Canceled)

1           21. (New) A method for on demand functional verification of a software  
2 model (40) of an application specific integrated circuit (ASIC), wherein said  
3 software model is written in a low-level programming language and separately  
4 causes a model of the circuit to be generated and debugging of functional verification  
5 tests to be applied to the model of the circuit for constituting a verification platform,  
6 comprising a transmission mode and a verification mode:

7           - creating in the transmission mode an autonomous circuit emulator (1),  
8 obtained by replacing the software model (40) which is in a low level programming  
9 language physically describing the circuit under design to be validated with a high  
10 level language abstract description generating response data structures in accordance  
11 with a functional specification (20) of the project as a function of the stimuli  
12 received, this mode being called the “transmission mode.”

13           - integrating the software model (40) in a verification mode into a  
14 verification platform, and connecting previously validated autonomous circuit  
15 emulator (1), in parallel, to interfaces of the software model (40) of the circuit, and to  
16 an environment emulator (11, 21, 22); and

17           - utilizing the verification platform as a reference for the validation of  
18 response data transmitted by the software model (40) of the circuit.

1           22. (New) A method according to claim 21, wherein a user:

2           • generates, using a data processing system, the autonomous  
3 circuit emulator (1) which provides a simulation configuration  
4 corresponding to the software model (40) of the ASIC using the functional  
5 specification (20),

6           • writes, from the functional specification (20), and stores in a  
7 test platform (21, 22, 23) for integrated circuit models, a program (51) for  
8 testing the software model (40) of the ASIC, comprising input stimuli  
9 sequences to be provided to the software model (40) of the ASIC, which

10           the autonomous simulation configuration (1), based on the functional  
11           specification (20), corresponds to output stimuli sequences,  
12               •     links together, and activates, the autonomous simulation  
13           configuration (1) and the test platform (21, 22, 23), and  
14               •     observes the output stimuli of the HDL-type model (40) of the  
15           ASIC in order to functionally validate the system constituted by the  
16           software model of the ASIC circuit and the validation test program (210),  
17           and thus validates the software model (40) in comparison to the functional  
18           specification (20) .

1           23. (New) A method according to claim 21, wherein the autonomous  
2           configuration (1) communicates with the user to control the activation of previously  
3           created and stored models of input stimuli sequences defined in a high-level  
4           programming language, and controls the activation of associated programs (90) for  
5           the progressive validation of test sequences determined from the models.

1           24. (New) A method according to claim 21, wherein a user writes and  
2           provides the functional specification (20) in a low-level programming language,  
3           specifying functional models of circuits.

1           25. (New) A method according to claim 22, wherein the user writes and  
2           provides the functional specification (20) in a low-level programming language,  
3           specifying functional models of circuits.

1           26. (New) A method according to claim 23, wherein a user writes and  
2           provides the functional specification (20) in a low-level programming language,  
3           specifying functional models of circuits.

1           27. (New) A method according to claim 21, wherein a user provides the  
2           functional specification (20), in the form of a program in a low level programming  
3           language of functional models of circuits, and a program in a high level  
4           programming language of functional models of circuits, and the user controls the

5 autonomous simulation configuration (1) so as to perform a co-simulation by  
6 synchronizing the execution of the two specification programs.

1 28. (New) A method according to claim 27 wherein the low level language  
2 is an HDL type and the high level language C<sup>++</sup>.

1 29. (New) A method according to claim 21 wherein the verification  
2 platform verifies that the responses of the software model of the ASIC are within  
3 response time ranges specified in the functional specification (20).

1 30. (New) A method according to claim 22 wherein the verification  
2 platform verifies that the responses of the software model of the ASIC are within  
3 response time ranges specified in the functional specification (20).

1 31. (New) A method according to claim 23 wherein the verification  
2 platform verifies that the responses of the software model of the ASIC are within  
3 response time ranges specified in the functional specification (20).

1 32. (New) A verification platform for on demand verification of a software  
2 model of an integrated circuit (ASIC), characterized in that it comprises data  
3 processing means that allow a client to select test models producing input stimuli for  
4 the ASIC, said data processing means being constructed and arranged to read  
5 functional specification elements (20) of the ASIC and comprising programs (90)  
6 that form an emulator and generate a functional validation test program (51)  
7 constituted by output stimuli, from the input stimuli and the functional specification  
8 elements (20).

1 33. (New) A verification platform according to claim 32, comprising a  
2 library of functional models of circuit blocks for ASICs and means for selecting  
3 models through a definition file of the integrated circuit configuration, in order to  
4 create a model corresponding to the functional specification of the ASIC that is  
5 integrated into the definition of its environment.

1       34. (New) A verification platform according to claim 32, further including, in  
2       a link connecting the platform to a client, two serial programming language  
3       adaptation circuits (11, 12), that transform commands in a high level programming  
4       language (C<sup>++</sup>), used by the client, into commands in a low level programming  
5       language used by the ASIC model, and respectively, to the commands in the low  
6       level programming language back into commands in a high level programming  
7       language.

1       35. (New) A verification platform according to claim 32, characterized in  
2       that the platform includes means (90, 10) for executing operations at the same time  
3       as the simulation, and upon detection of an error interrupting operations at the very  
4       moment the error appears.

1       36. (New) A verification platform according to claim 32, characterized in  
2       that the functional specification elements (20) are constituted by a truth table  
3       corresponding to the functions of the various functional circuit elements of the ASIC  
4       software model (40), and a propagation delay to be respected between each input and  
5       each output.

1       37. (New) A verification platform according to claim 32, characterized in  
2       that the functional specification elements (20) are constituted by a behavior table  
3       corresponding to the functions of the various functional circuit elements of the ASIC  
4       software model (40), and a propagation delay to be respected between each input and  
5       each output.

1       38. (New) A verification platform according to claim 32, further including a  
2       cache memory (962) for storing the blocks used by nodes according to node  
3       addresses, and means for managing, for an address used by one or more nodes, a  
4       presence vector with one presence indicator per node.

1        39. (New) A verification platform according to claim 38 characterized in that  
2 the programs (90) are object-oriented and the emulator is structured as a set of  
3 classes for managing a collection of execution hypotheses for a transaction in a  
4 memory block of the software model, and for managing transactions that are  
5 concurrently colliding using the same memory block.

1        40. (New) A verification platform according to claim 38, characterized in  
2 that algorithms of the programs (90) of the emulator perform the following functions:  
3 generating predictions, eliminating predictions, readjusting incorrect predictions,  
4 reducing the number of valid hypotheses, and terminating collisions

1        41 (New). A verification platform according to claim 40, characterized in  
2 that the emulator of the circuit generates predictions without having to obtain  
3 additional information on the internal operation of the circuit under design.

1        42. (New) A verification platform according to claim 38, characterized in  
2 that the platform is used as an emulator of a router circuit, a circuit with cache or a  
3 router circuit with cache.

1        43. (New) A verification platform according to claim 38 for testing a  
2 software model of an integrated circuit (ASIC) on demand characterized in that the  
3 platform comprises an ASIC emulator (1) for controlling a comparator (23) that  
4 receives values generated by a software model of the ASIC circuit tested, upon  
5 reception of stimuli sent by at least one stimuli generating circuit (21) storing a test  
6 program, an interface (11) for translating the stimuli from an advanced language into  
7 a low level language corresponding to that of the software model, and means for  
8 validating the verification in case of the detection of a collision by the comparator  
9 (23).

1        44. (New) A verification platform according to claim 38, further comprising  
2 means for selecting the response to stimuli that depend on the composition of the  
3 circuits tested, said means for selecting being constituted by a model generated by

4 means for selecting functional models from a library, which associates with each of  
5 the models the responses to a given stimulus, the model corresponding to the  
6 composition of the circuit to be tested.

1 45. (New) A verification platform according to claim 44, further including  
2 means (7) for storing responses selected so as to create a test model (70) to be  
3 applied to the

1 46. (New) A verification platform according to claim 32, characterized in  
2 that each transaction is constituted, at the level of each interface, by a request packet  
3 and one or more associated response packets, wherein the values of the parameters  
4 and/or the transmission time constraints of the packets can be forced from the  
5 functional test program executed by the emulator of the environment, which  
6 appropriately translates all of said parameters during the transmission of the packets  
7 to the terminals of the software model of the design.

1 47. (New) A verification platform according to claim 43 characterized in  
2 that each transaction is constituted, at the level of each interface, by a request packet  
3 and one or more associated response packets, wherein the values of the parameters  
4 and/or the transmission time constraints of the packets can be forced from the  
5 functional test program executed by the emulator of the environment, which  
6 appropriately translates all of said parameters during the transmission of the packets  
7 to the terminals of the software model of the design.